Downloaded from orbit.dtu.dk on: May 05, 2023

DTU Library

Computer Installation. Annual report 1 Januar - 31 December 1978

Risø National Laboratory, Roskilde

Publication date:
1979

Document Version
Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Citation (APA):
Risø National Laboratory, R. (1979). *Computer Installation. Annual report 1 Januar - 31 December 1978*. Risø National Laboratory. Risø-M No. 2177

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

RISØ-M-2177

ANNUAL REPORT 1 JANUARY - 31 DECEMBER 1978
COMPUTER INSTALLATION

Abstract. This report describes selected parts of the activities at the Computer Installation of Risø National Laboratory. Information given may be preliminary.

UDC 061.62 : 621.039 (489) Risø National Laboratory
UDC 681.3

May 1979
Risø National Laboratory, DK 4000 Roskilde, Denmark

ISBN 87-550-0607-8
USSN 0418-6435

Risø Repro 1979

CONTENTS

1. THE ROLE OF THE COMPUTER INSTALLATION AT RISØ NATIONAL LABORATORY

The Annual Report of Risø takes a new form this year, being a combine of reports from the individual scientific departments and assisting installations. This means that the Computer Installation for the first time describes its activities in a comprehensive annual report in English.

The bulk of this report is a number of papers on selected projects that the Computer Installation has been working on during 1978. As this report will reach many readers who are not familiar with the Computer Installation at Risø, we have however decided to give a general introduction in this first paper.

1.1. History

The first Danish computer - the only one-of-its-kind DASK - became operational in late 1957. It was built and run by Regnecentralen, at that time an institute under the Danish Academy of Technical Sciences. Risø had its first three computer codes, models in reactor physics, running on DASK in the summer 1958. Risø's use of DASK continued for some years, but in 1962 a GIER-computer was installed in the Reactor Physics Section at Risø, and it took over the total work load.

GIER was a transistorized computer with a ferrite-core of 1024 42-bit words and a magnetic drum with 12,800 words. It was manufactured by Regnecentralen which sold about 50 of them. One of its merits was the Algol-compiler.

In the late 60's the extended use of edp at Risø exceeded the capacity of the GIER. An increasing part of the computations were performed on an IBM 7090, later IBM 7094, at NEUCC, the Northern Europe University Computing Centre, which is situated at the Technical University of Denmark at Lyngby, 10 km north of Copenhagen.

In November 1970 a Burroughs B6500 was installed at Risø. By later additions it has evolved into our present B6700, the configuration of which is given in Appendix A. Risø's use of NEUCC dwindled quickly. During 1977 and 1978 use of external capacity has again reached a significant size, partially due to a delayed extension of the B6700, partially because external cooperation in some instances dictated the use of another brand of computer.

1.2. Organization

The Computer Installation functions as an internal service bureau at Risø. Its head reports to Risø's Board of Directors. All departments of Risø are entitled to nominate an edp-representative, all of which meet twice yearly to approve reports on the period's work and discuss plans for the future. Furthermore the Directors have appointed an advisory edp-committee of six persons (nominated by the representatives), who work on the coordination and planning of edp-investments.

The personnel of the Computer Installation in 1978 is listed in Appendix B. The permanent staff consists of five graduates, four computer operators, three programmers and two secretaries. The work of this staff is concentrated around operating the computer and providing expertise in the fields of numerical mathematics, statistics, operation research, systems software, and software engineering.

This expertise is available for the users through courses, colloquia,and private consultations. Now and then the users' problems in the fields mentioned lead to research or development of methods to solve these problems. Several papers in this report describe such work.

On the other hand the Computer Installation does not work as a programming bureau except in exceptional cases. The bulk of the application programming at Risø is performed in the users' departments and is often a tightly integrated function in the research work.

1.3. Hardware and Software

The Burroughs B6700 computer has an architecture which is rather unusual. It is a stack-machine which makes it very suitable for work in high-level languages as algol, fortran, and cobol. In fact there is no assembler because the algol-language has been extended with the necessary facilities for bit manipulation, etc. Furthermore, there is a virtual memory system, which automatically segments all programs at the block/subroutine/paragraph level. Addressing of files is by their name, e.g. a magnetic tape is normally identified by its contents and not by the station where it is mounted. These features mean that many details from "the art of programming" are not necessary for application programmers, a fact which lies behind the strategy of leaving the programming work to the end users.

The users may work with the computer through the peripherals (card reader, line printer etc.) in the computer room, through a "Remote Job Entry" station at the other "geographical end" of Risø, or through about 30 dial-up terminals, mainly TTY-compatible Visual Display Units, which are distributed in the departments of Risø.

Also a few of Risø's many PDP-minicomputers are equipped with modems and software and may act as a B6700-terminal. This may be used to transmit data, which the mini has logged from an on-line experiment or as a method of providing more computer power for the user of the mini.

One PDP-8A is placed adjacent to the B6700 and is used to read and write floppy disks. Paper tape equipment is in the process of being implemented on this PDP-8, and magnetic tape cartridges are being considered for implementation, thereby collecting the handling of media for data logging on a self-service unit. (There are a paper tape reader and puncher in the computer room).

It should be noted that we have no real-time on-line connections to the B6700. The experiments at Risø are typically controlled by a mini which later may deliver data to B6700 by datatransmission or by physical transport of data carrying media. We thereby avoid some constraints on the operation of the computer.

To finish the discussion of terminals, the B6700 may itself act as a terminal, simulating an IBM 2780 versus an IBM 370/165 at NEUCC or using an NTR-protocol versus a UNIVAC 1100/82 at the Regional Edp-Centre at the University of Copenhagen (RECKU). The B6700 software for these functions has been developed at Risø. Finally two Calcomp digital plotters are connected to B6700 via locally built electronic interfaces which act as modems towards the computer and deliver the proper bits to the plotters.

By far the largest part of the programming effort is done at the interactive terminals. The "Cande" software of the B6700 permits the users to edit their files, run interactive programs, and enter jobs into the queues for batch-processing. Each user is permitted to occupy a certain space on one of the disk packs while the system software is on another pack and the two last disk drives are used to mount user-owned packs on request.

The system software is as far as possible the vendor's standard software in order to avoid local maintenance problems. Locally developed system software includes the Risø Computer Library, programs for file-handling and back-up, the plotter system, logging and accounting programs and specific request sets for the data transmission protocols in use. Some of these items are discussed in other papers in this report.

1.4. Use of the B6700

Appendix C gives some statistics on the use of B6700 in 1978. The computer is scheduled to run in 3 shifts, 5 days a week, minus the time reserved for preventive maintenance. The night shift is shortened or the weekend included as need be.

This is not the place to report on the problems which are solved by means of the computer. That lies within the user's area.

Our aim here is to give an indication of the type of problems which are run.

They fall into two main groups. Approximately 60% of the capacity is used for development and use of models, about 40% is used for the processing of experimental data and about 3% is taken up by administrative tasks like inventory and budget control.

The models are concerned with technical and scientific problems, e.g. phenomena in solid state physics, safety of nuclear reactors, load on and stresses in mechanical structures like reactor containments, wings on windmills, and steel chimneys, behaviour of nuclear fuel elements, and deposition of radioactive fall-out after a hypothetical accident. Typically, all tests and use of these models are very demanding on processor time but do not need much input and output.

Many experiments at Risø produce a large number of data, which later are brought to the computer for reduction and analysis. Some experiments still use punched paper tape as a medium for the registration and transport of the data, but magnetic tape is coming into greater and greater use, and they of course permit a much larger number of data to be handled. A few experiments have floppy disks and magnetic cartridge tapes are expected shortly. Finally data may be sent from a minicomputer which is dedicated to an experiment, over the telephone lines to the B6700.

Important users in this group are the meteorological section, the nuclear geophysicists' airborne survey of uranium deposits in Greenland and their measurements on drill tests, and physical and chemical laboratory experiments.

Throughout the year the staff of the Computer Installation has had an extensive cooperation with the users. We believe that this has resulted in an offer of hardware and software facilities, a daily operation and a specialized support which is a valuable help in reaching the overall goals of the Risø National Laboratory. Important examples of the year's work are described in the following papers.

2. EDP-SYSTEM PERFORMANCE MEASUREMENT AND OPTIMIZATION

This paper attempts to describe some of the methods applied at Risø's B6700 computer in measuring and optimizing the performance of the system. The methods are numerous and very different and the present paper will deal with only some of the most important ones, and such ones that can be understood without specific knowledge of the B6700 system.

2.1. Objects for Optimization

Many different areas within the system may be objects for optimization, the most important ones are:

2.1.1. Hardware Configuration

In a system composed of a number of more or less independent units you often have a possibility of identifying one of these units as the bottleneck of the system. An important task in optimizing the performance of a system is to identify such bottlenecks and eliminating them either by installing more hardware or by moving the load to other less heavily used units.

2.1.2. System Software

The system software is the collection of programs that controls the operation of the system. Most of these programs are constructed and maintained by the vendor, and they are often very large and complicated. Thus, in most cases, the installation must trust these programs, without really having knowledge of their detailed function.

The system software has often a number of parameters used for the adaptation of the software to the actual hardware configuration and the actual user environment.

2.1.3. Application Software

The application software is constituted by the programs developed to solve specific user problems. These programs do vary very much in size as well as quality. However, as these programs are responsible for a very large part of the system utilisation, it is of major importance to focus on their optimization. Also in the case of application programs it is often possible to identify a few of these as system bottlenecks, and furthermore it may be possible within such programs to isolate parts of the program as a bottleneck.

2.1.4. Operation

The way in which the users are given access to the system may often influence the system performance drastically. However, when dealing with operations it is important to realize that the users and the operations staff are part of the system, and that it is this total system which is the object of the optimization. When this fact is taken into account, one will often become aware of a conflict between the internal performance of the system, measured as the number of useful operations executed, and the external performance of the system as experienced by the user.

2.2. Processes, Resources and Queues

A large edp-system may be thought of as a collection of processes (programs) competing for a number of resources (units) which may operate independently of the rest of the system for a given period. The resources of major interest with respect to performance optimization on the RISØ-B6700 system are:

a) 1 central processor
b) 1 input-output processor, with 6 channels
c) The core store
d) The peripherals, connected to the core store through the channels of the input-output processor
e) 1 Datacommunication processor with a number of terminals
f) Finally, in some aspects, the operations staff may be regarded as a system resource.

The competition for the system resources is organized in the following way:

A process requiring a resource will get it if it is available. Otherwise the process will enter a queue of processes waiting for that resource. Whenever a resource becomes available, the operating system will check if anybody is waiting for it, and give it to the process at the head of the queue.

This queuing mechanism is the key to one of the most important methods for locating system bottlenecks. The number of processes which on the average are queued for a certain resource, directly indicates to which extent that resource acts as a system bottleneck. Since the average length of a queue is easily monitored, we now have a good method for system monitoring.

2.3. Load and Balance

The average depth of the queues (or the accumulated waiting time) may be interpreted as a measure of the load on the different resources. With some caution this measure may be used to compare the load on different resources although these may be of a very different nature. A system in which all queues are nearly equal in length, is said to be in balance in the sense that no single resource can be identified as a system bottleneck.

A system in which all queues are permanently empty is said to be underloaded, while a system in which all queues are permanently long is said to be overloaded.

An underloaded system does not utilize the resources completely, but provides a good service to its users. Neither does an overloaded system utilize the resources very well, since a process waiting in a queue often holds other resources than the one it is waiting for. The service provided by an overloaded system is generally bad.

A system out of balance may be balanced in different ways:

a) By extending the hardware in adding more of the scarce resource.
b) By substituting less loaded or cheaper resources for the scarce resource.
c) By establishing resource restrictions for the scarce resource.

Point b especially is an important area for software optimization. In the following, three examples of such substitutions are given.

Example 1, Virtual store:

The B6700 system operates a virtual store, where the operating system moves some data* from core to disk in periods when these data are not needed in core, while they are moved back to core again when needed. In this way a user may have a larger area in core at his disposal while in return in certain situations he pays the cost of waiting for the transfer between disk and core.

The mechanism may be envisaged as two substitutions:

a) The cheaper disk-store is substituted for the more expensive core-store.
b) The time spent in transferring data between disk and core is substituted for space in core.

---

* "Data" is used in the full sense of the word, i.e. the program is also data.

The latter substitution may imply a risk of establishing a bottleneck in the system, if the number of disk transports exceeds the capacity of the disk channel.

Example 2, Storage of files:

At the B6700 system most files are stored on a system disk. An extension of the disk to a size large enough to hold all Risø's files would be unacceptable from an economic point of view. A back up system (described in more detail in Chapter 3 in this annual report) has therefore been set up, to transfer files from disk to tape and vice versa. In this example the cheaper tape is substituted for the more expensive disk. A positive side effect of this substitution is that files are protected against damage on the disk, while the delay involved in the reloading process is a negative side effect.

Example 3, File placement:

The B6700 disk store is configured as a small heat-per-track-disk with fixed reading heads and fast access, and a number of larger disk packs with a slower access. The units may operate simultaneously and may to a great extent substitute for one another. However, a lot of other considerations have to be taken into account when deciding on which files are to be placed on either, i.e. which substitutions are useful:

a) The packs are mountable while the disk is fixed.
b) The risk of damage to the data is much higher on disk than on pack.
c) From an operational point of view logically related files should be kept together on the same physical unit.
d) The data should be equally distributed among the units.
e) The data transports should be equally distributed among the units.
f) The total solution must appear simple and straightforward to the user.

As may been seen from the above, such a substitution must observe a series of constraints, which normally will not be equally well fulfilled. Compromises are often extremely difficult to establish.

2.4. Operational Stability

An important cause of inefficiency in an edp-system is the interrupts of operation, whether these are caused by hardware or by software. It is therefore very important to keep the number of such interrupts as low as possible. The best way to do this is to perform a detailed analysis of all such interrupt situations, and in cooperation with the vendor decide what can be done to avoid the error in the future.

2.5. Summary

Performance measurement and optimization is a continuous activity covering hardware, software and the human environment of the system. It requires a knowledge of the whole system in the broadest sense of the word "system". In practice it is done by a sequence of isolated improvements, but it is very important to keep in mind the effects on the total system, so that no local optimization causes a global or system wide pessimization. The effect of major modifications should as far as possible be verified by measurements on the entire system.

3. FILE-HANDLING POLICY ON B6700 AT RISØ

The objectives of the file handling system are to

- support	a "virtual-disk-storage" system
- provide	backup facilities for security, recovery, and archiving
- define	user and installation responsibilities
- produce	fast and safe recovery of faults
- interface	users and operational working procedures
- insure	system integrity at any time
- minimize	risk of losing files
- optimize	utilization of available resources
- guarantee	working space throughout the day
- report	status and actions taken
- balance	input/output operations between physical units.

The use of the B6700 Cataloging operating system in interaction with local utilities forms the fundamental basis of fulfilling these objectives.

A high level of automation has been implemented in well-defined operational procedures; still ongoing efforts are spent in maintaining system integrity through monitoring, debugging, performance measurements and improvements, and performing recovery after software and hardware failures.

3.1. Virtual-Disk-Storage

The disk storage configuration consists of one head-per-track disk (labelled DISK) and four diskpack-drives.

Two diskpack drives are available for mounting packs on user's requests or for special purpose packs, and two diskpack drives are reserved for stationary mounted operational packs labelled PACK and USERPACK.

PACK	- contains System software (compilers and utilities). Application software (mathematical and statistical symbolic procedures). Temporary system files (log-file, batch-load and definition of legal users). Temporary userfiles. (Short life files, typically a few rows).
USERPACK	- contains Permanent user-files.
DISK	- contains Files used by the operating system (permanent read-only codefiles and temporary virtual-memory-overlays and datacommunication tank-files). DISK is inaccessible to users.

The linkage between these disk-volumes is defined by a family substitution specification:

FAMILY DISK = USERPACK OTHERWISE PACK

with the following semantics·

All files declared to be on DISK are directed to USERPACK. On output the file is unconditionally placed on USERPACK. On input the file is looked for on PACK, if it was not found on USERPACK (all software is read only and is found on PACK, but without involving special actions from users).

In this way userfiles and system software is distinctly separated on different volumes, and input/output operations are distributed to different sources and destinations, attempting to balance input/output performance between physical units.

The state of files on USERPACK and PACK are clearly different. Files on PACK tend to be in a permanent and stable state, whereas files on USERPACK are continuously updated, and the load of files grows with new files created throughout the day.

With a limited resource of disk-space on USERPACK (shared by all users), the growth is controlled by the implementation of "virtual-disk-storage". In a true virtual system files with lowest access frequency are placed in secondary storage on magnetic tapes, while files with highest access frequency are kept resident in primary storage on disk. But other considerations have to be taken account of so that no single user can cause removal of all other userfiles by overloading the disk with new files.

The criteria for moving files to secondary storage become somewhat more complex. Files are moved if one of the following is valid:

- Last access to the file is older than 14 days (low access frequency).
- The user has exceeded the maximum share per user (2% of the total size). This user's largest files are removed from primary storage without consideration of access frequency. That should minimize the number of reloads, tape mountings, and tape spooling, but it is likely to remove files with highest access-frequency. Therefore, the files accessed within the previous 24 hours are given priority by reduction in size calculations.
- The total available working space is less than 25% of the total size. Files with largest product of size x (days since last access + 1) are removed to cut down the load of files to 75% of total size, insuring working space for some time.

Files are reloaded from secondary storage on user's request.

Presently this virtual disk system contains approximately 25000 files with 2000 files resident in primary storage, and the secondary storage is 1000 magnetic tapes holding a various number of files.

Additional disk-storage is available to users with large amounts of data on special packs, which are mounted on user's request.

These packs have separate virtual-storage but administration of the resources reside within the user's responsibility.

3.2. Backup Facilities

After each day of work the file manager utility is run during night shift.

All new files created and all files updated on USERPACK since the last run are backed up on magnetic tapes. After completion of this run these files exist in two copies - one copy in primary storage and one backup copy in secondary storage.

When a file has not been updated for a week (which means that it is in a stable state) or when a file is removed from primary storage by the utility, a second backup copy is produced. The file then still exists in at least two copies - two copies in secondary storage and one copy in primary storage, if the file was not removed. The system catalog file is backed up after all other backup, and will reflect the current status of all files, immediately after completion of the file manager run.

The backup copies in secondary storage on magnetic tapes are the file archive, from which the files may be reloaded on user's request or in conjunction with recovery of a damaged pack.

Once each week the file manager utility initiates a pack-to-pack backup of the current contents of USERPACK to a special pack named BUPACK. This backup pack may then be relabelled to substitute USERPACK in case the current USERPACK is damaged beyond repair. BUPACK is used only for fast recovery purpose with a maximum loss of one week's updates. The current status of files is however kept intact in the system catalog file, so that all files can be retrieved from the backup tapes of the last week. The maximum loss of updates is then reduced to one day's work, i.e. hours since last run of file manager utility.

Furthermore, a special pack named FIREPACK is mounted the last three working days of each month. Users may then copy files which are subject to special security concerns to FIREPACK. At the end of the month, contents of FIREPACK is backed up, on magnetic tapes, which are placed in a storage in a building separate from the computer installation.

Backup of files on PACK reflect the stable state of the software files. Three tapes are used - one is backup tape for system software and utilities, and two backup tapes contain all application software (primarily symbolic routines). Recovery of a damaged PACK is performed by loading all files from these three tapes, and adding the system catalog file from latest backup. The cataloging system will rebuild the catalog file to update all status changes on all disk volumes.

Files on user owned packs are backed up on user's request. No automation is involved in managing files and working space on these packs. All actions must be initiated by the user.

3.3. User and Installation Responsibilities

As described previously, all files placed by users on USERPACK are backed up and entered in the disk storage system, without further actions from the user.

No file is automatically deleted from the system. Deletion is user's responsibility, though the installation does encourage users to clean up by deleting outdated files every now and then. Such encouragements are announced when the amounts of file informations tend to grow rapidly and causes disk space extensions to the system catalog file, permanently using extra disk resources.

The installation is responsible of administration of overall system resources, while the user is responsible for the extent of the burden he imposes on the system. As an example, the available working space may be used up during the day, as the assumed 25% of the total size is not a foolproof estimate to guarantee working space throughout the day. On the other hand it has proved to be satisfactory, but occasionally a few users impose heavy loads on the system.

An announcement of the overload will usually cause the users to relieve the situation by removing files to avoid extra runs of the file manager utility, which involves closing the user access to the system as no work can be performed until working space is available. The users are aware of their responsibility.

The installation is not committed to keep track of generations of files. There is always only one file with a specific title and only the latest version is accepted by the file manager.

The cataloging system does however keep track of the two latest versions of any file, which may be useful for recovery when a user finds out his file is corrupted.

Users who wish to keep track of more than one generation of files must use different titles of the different generations to distinguish among them.

Each user has a separate directory of files, as the usercode is the first item of the file title. The usercode is the key to legal access to the files, and is protected by a secret password defined in a system-file containing additional information of ownership and charge codes. However, the user may allow other users to access files in his own directory by specifying public security attributes on selected files.

3.4. File Status Reports

The interface between users and installation is heavily dependent on the users' access to file status informations, which are necessary to request the installation to reload the files.

Status reports may be obtained by running a local reporter utility which produces a printed list of all files in the user's directory. This utility may even be run or started as batch job from an interactive terminal.

Or the user may get file information directly to the terminal by a file status request for a specific file, a subdirectory, or even for all files in the user's directory on any specified disk volume.

Operational staff is informed of this by daily reports. The file manager utility reports the current status of all files resident in primary storage, and all actions performed on these files. Furthermore a list of all tape volumes, which are referenced as back up tapes, is produced sorted by number of references.

An option of the file manager utility is to produce a report on all tape volumes sorted by serial number with lists of references per usercode to each backup tape.

3.5. Management of Secondary Storage

Reports on the status of backup tape volumes are used to find candidates to be released and reused as a scratch tape.

When new versions of files are created, the old backup references in the catalogue are outdated, and the number of references to backup volumes are continuously descending. Eventually some of the backup volumes will have only a few references.

The tape release utility will read the tape directory and initiate a loading of all files with valid references to the tape, before deleting the references. After completion of this run the tape is released and may be reused and the files from the tape are resident in primary storage without backup references. During the next run of the file manager utility new backup of these files will be provided, as if they were new files created during the day.

This is the way to control growth of secondary storage, and it is performed during night shift to make it transparent to the users.

System integrity suffers injury when a backup volume is damaged, because files with backup references to the volume may be inaccessible. Recovery can be performed in two ways. If all files with backup reference to the damaged tape volume have alternative backup references or are resident in primary storage, the files may be retrieved from alternative backup, before references to the damaged volume are deleted. If this recovery is not possible because no alternative backup exists, a library tape recovery utility can be run to attempt to reconstruct the damaged volume. Very few files have been lost, and then still with a maximum loss of one day's work, as the files have been recreated or retrieved from the preceeding generation.

3.6. General Comments

The load of files form part of the research environment with a mixture of program development with minor sets of test data, and heavy production runs with larger amounts of data. Presently there are no online or database systems which would require auditing and special considerations in file management.

An auto load function (i.e. load on reference) has been considered, but would be unacceptable while the secondary storage is on 800 BPI-tapes and with a slow moving, old-fashioned magnetic tapecluster; in any case the amounts of files and tapes would prevent implementation. The files must be preloaded as part of preparation of processing.

Other features are considered to improve the file handling management and suggestions for improvements in the B6700 system software have been forwarded to Burroughs. Improvements to the system software are currently implemented and maintained locally. Some remove inexpediencies and some correct bugs. Other users of Burroughs equipment have experienced the same major problems in the system software, but still have no solutions. Generally our system integrity is the most consolidated within the scope of Burroughs users of cataloging operating system. Presently we know of no major software problems which we are unable to control.

4. PRODUCTION OF BRAILLE VIA EDP.

4.1. Automatic Production of Information for the Blind.

Still bigger amounts of information are nowadays brought into machine-readable form, i.e. a form which can be read directly by a computer. This holds true for fiction as well as special literature, since modern photo composition implies that a text is brought into machine-readable form. It also holds true for much illustrations stuff. Note stuff (music) is sometimes stored in machine-readable form.

With the machine-readable form as starting point, the information may be printed in different forms. As an example, machine-readable text may be printed in large type, which can be read by the weak-sighted, or in Braille which can be read by the blind. The only requirement is that the computer is equipped with a unit, which can print characters of the kind desired.

Such an automatic production of Braille from machine-readable text is obviously much faster and cheaper than production by old fashioned methods, where a person copies the original test on a "Braille-keyboard". The automatic production from machine readable stuff opens up wide perspectives for blind and weak-sighted persons, whose supply of information is catastrophically small, even in the richer and highly developed countries.

Today the blind in Denmark have access to about 2% of the fiction that sighted persons have access to. The publishers have, however, in the last few years shown increasing interest for recording books on cassette tapes. This is of course of great importance, not only for the blind, but also for the word blind and other reading handicapped persons.

In Denmark the blind have practically no access to information about local politics. Local newspapers on cassette tapes however have begun to appear in recent years.

As regards fiction and local newspapers, cassette tapes probably represent the best solution. However, there are also serious difficulties in providing the necessary special material for the blind having a job or being under education. It may be in this field that the automatic production of Braille is of greatest interest.

4.2. Braille Writing

Braille writing was invented by the Frenchman Luis Braille in 1825. It is used today by the blind all over the world. In Braille writing each character is printed as a combination of raised dots on the paper, and it is a matter of common knowledge that the blind read a text by sensing the dot combinations with their finger tips. Each character occupies a so-called Braille cell. The cell has six dot positions (Fig. 1), and thus 2**6 = 64 different characters may be formed. A little less than half of these are of course used to represent the letters of the alphabet. Other characters represent punctuation signs.

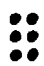


Fig. 1. Braille Cell

The digits 1-9 and 0 are represented by the same character as the letters A-I and J, in which case the letters are preceded by a special Braille character, the so-called numeral sign. The numeral sign has effect till the first following space.

In many languages, among them Danish, a shorthand system has been developed. The unused Braille characters are used as abbreviations for syllables or parts of syllables which occur frequently in the language. Of course, this implies the dis-anvantage that different languages have different abbreviation systems. On the other hand the use of abbreviations makes the material less voluminous and speeds up reading.

Notes may be represented by Braille characters. A note system was developed already by Luis Braille. It is fundamentally his system that is used today. The system has the advantage of being internationally accepted.

4.3. EBCDIC-Braille

When computer programs are listed in Braille, the usual Braille writing cannot be used. A special Braille writing (in the following called EBCDIC-Braille) must be used instead. In algol and fortran programs an identifier name may be formed as an arbitrary combination of letters and digits, and this obviously infers that abbreviations cannot be used. In usual Braille the number seven is written as the letter G preceded by a numeral sign. If a program declares the two identifiers A77 and A7G, it is not sufficient just to introduce the numeral sign after the A, because this will give the two names exactly the same appearance. The only possible way as far as program listings are concerned, is to redefine the Braille writing in such a way that a one to one correspondence is established between EBCDIC characters and Braille characters.

Unfortunately, an internationally accepted Braille representation of the EBCDIC character-set does not exist. About ten different representations have already been developed around the world. One of these representations was developed at RECKU (The Regional Edp-Center at the University of Copenhagen). For the letters in the English alphabet this representation uses the same Braille characters as usual Braille. For the digits 1-9 and 0, the lowered letters A-I and J are used, while the numeral sign is omitted. (The first ten letters of the alphabet only use the four topmost dot positions of the Braille cell, and may therefore all be lowered). Digits are thus represented in a way which is close to usual Braille. Since usual Braille uses the lowered letters as punctuation signs, most of the punctuation signs must be redefined. The one to one correspondence between EBCDIC characters and Braille characters means that EBCDIC-Braille is a simple six-bit character representation on a par with, for instance, BCD.

4.4. Hardware

Today a number of units which print Braille characters instead of normal characters can be attached to a computer. For instance a number of "Braille-terminals" exist. In Denmark a single installation is available today. This installation was bought by the Danish Society of the Blind. It has been placed at the State Library for the Blind, and is put at the Library's disposal. One end of the installation is a lineprinter which has been rebuilt to print Braille characters instead of normal characters. The other end is a magnetic tape unit. The two units are connected by a minicomputer. In principle we can now use any computer to transform machine-readable text into Braille. The result is written on a magnetic tape which is mounted on the tape unit of the installation mentioned by means of which the text is printed in Braille.

4.5. Software

The transfer of a machine-readable text to Braille is not just a simple transfer from one medium to another. The text must on the way be rearranged by a computer program, in the following called a Braille processor. The problems to be treated by a Braille processor can be roughly devided into two groups, namely, introduction of abbreviations in the text and editing problems. The introduction of abbreviations is a difficult problem for the computer due to the ambiguous structure of the language, and the illogical way of dividing words into syllables.

In some cases the problem is in a way insoluble. As an example the Danish word "vandring" may mean two different things, namely, a water-ring, where the division in syllables is vand-ring, or a walk where the division in syllables is van-dring (derived from the verb "vandre"). The different division in syllables implies the use of different abbreviation symbols. Whether the word occurs in one sense or the other, i.e. whether it is to be abbreviated in one way or the other, can only be decided by considering the surrounding text. Such a context analysis cannot be mastered by a computer. It is however possible to develop Braille processors, which introduce abbreviations correctly in by far the major part of a text. A fully acceptable result is for instance supplied by the Braille processor, which the Danish Society of the Blind had produced in relation to the buying of the above-mentioned installation.

Machine-readable text from photo composition systems contain a number of graphical codes, which determine the arrangement of the printed text, the font etc. The graphical codes differ from system to system, and as part of the editing process it is therefore necessary to code a preprocessor for each system that is going to be used. The major task of the preprocessor is to transform the graphical codes into a set of Braille codes that is a set of editing codes being known by the Braille processor.

Usually it is necessary to make up the original text. Firstly, Braille writing allows a linewidth of only 30-40 characters, whereas normal printing generally uses a linewidth of 70-90 characters. On the other hand the introduction of abbreviations involves a reduction of the original linewidth. Also the editing phase contains problems that are nearly insoluble for the computer, e.g. the computer should detect quotations in foreign languages. Firstly, Danish Braille demands that a quotation in a foreign language is preceded and succeeded by a special Braille character. Secondly, the introduction of abbreviations must be "cut out" (using the abbreviation system of one language in another language leads to a confusing result and therefore, the rule has been introduced that abbreviations may not be used in quotations in foreign languages).

4.6. The Work at Risø

Risø has done some work in the field of Braille production, due to the fact that one of the employees of the Computer Installation went blind a couple of years ago. The Braille processor (RISOE/BRAILLE) has been developed step by step as new demands to the program arose. The first demand was program listings in Braille. The Braille representation of the EBCDIC-characterset, which was introduced at RECKU (confer section 4.3), is used at Risø too.

The handling of card images in RISOE/BRAILLE is controlled by a set of options. One of these options determine whether the leading blanks are removed or not. Another option determines whether the string of blanks between the statement and the sequence number is removed or not. A third option decides whether the sequence number is included in the listing, and a fourth option may be used to move the sequence number to the front of the card image. The string of characters that is left by these operations is cut into pieces of thirty character length.

Each piece is translated into EBCDIC-Braille (character to character translation) and is written on magnetic tape as a finished Braille line. This magnetic tape is sent to the State Library for the Blind, where the Braille printing is done on the installation that was described in section 4.4. In this way it is a simple matter to produce program listing in Braille. The removal of superfluous blanks is very important, because Braille-output is essentially more voluminous than the corresponding output in normal print.

The next demand was that the blind employee should be able to read himself what he published in normal print. In other words the demand is that the same text should be printed in Braille as well as normal print. Instead of writing the text on a typewriter it is written to disk by means of a terminal. The text is stored on disk as card images, but only the first thirty characters are used in each image, i.e. a linewidth of thirty characters is used.

Of course it was a simple matter to produce a program which prints the card images on the lineprinter. This program writes two columns on each page so that a reasonable utilization of the paper is obtained in spite of the small linewidth. The RISOE/BRAILLE program already established, could of course be used on these card images as well as on card images that contain program text (version in Braille). Of course, the Braille version is without abbreviations, and it is written in EBCDIC-Braille instead of usual Braille. However, the difference between EBCDIC-Braille and usual Braille is not too big (confer section 4.3) and experience has shown that it is not very difficult for a person to convert from one type of Braille to the other. The system has the advantage that the Braille versions is an exact copy of the version in normal print, so that the blind employee has complete control over his publications in normal print.

The last demand so far is Braille versions of the most important manuals. Most system programs on a B6700 computer are written in algol and it was therefore first and foremost the Algol Reference Manual that should be transferred into Braille. Burroughs Corporation maintains some of the B6700 Manuals in machine-readable form, but unfortunately, the Algol Manual was not among them. The blind employee succeeded in obtaining a grant from the State Library for the Blind, so that the algol manual could be punched on cards. Among the manuals which Burroughs Corporation maintains in machine-readable form is the Input/Output Subsystem Reference Manual. Also this manual is important for the system programmer. It was therefore clear that RISOE/BRAILLE had to be developed further so that it is made able to handle machine-readable texts which were not produced with reference to Braille production.

As understood from the above it is a complicated affair to transform an arbitrary machine readable text into completely correct usual Braille, and there was not enough time and manpower for such a project. It was given up to introduce abbreviations in the text (moreover the manuals are in English language, so that the Danish abbreviations system would be irrelevant). Besides, it was decided to use EBCDIC-Braille instead of usual Braille.

However, a good deal of work was done with respect to editing. Among other things the original text is made up in such a way that the original paragraph structure is conserved. The most important of the options which have been introduced in RISOE/BRAILLE for this purpose are called oldmargin, newmargin, submargin, and linew. Oldmargin is the margin in the original text, i.e. the number of leading blanks in a text line. Newmargin is the margin in the Braille version. In case of a text with normal paragraph structure i.e. when the first line of a paragraph has more leading blanks than the succeeding lines, submargin is given a positive value, e.g. +2.

When RISOE/BRAILLE meets an input line with a number of leading blanks that is equal to oldmargin the leading blanks are first removed, and thereafter the line is cut into pieces of the length linew-newmargin. Each piece is translated into EBCDIC-Braille, and printed as a Braille line with newmargin leading blanks. If the program meets an input line with a number of leading blanks exceeding oldmargin, the line is again cut into pieces, but the first piece is given the length linew-(newmargin + submargin), whereas the other pieces are again given the length linew-newmargin. The first piece is printed as a Braille line with newmargin+submargin leading blanks, whereas the other pieces are printed with newmargin leading blanks as before.

This process conserves the original paragraph structure. Each paragraph in the original text is converted into a Braille paragraph in which the first line has newmargin+submargin leading blanks and the succeeding lines newmargin leading blanks. All lines have a total length of linew.

The algol manual contains Backus-Naur formulas of the structure

```
<unsigned number> :: = <decimal number>|
                       <exponent part>|
                       <decimal number> <exponent part>
```

Such a formula is looked upon as a paragraph whose first line has fewer leading blanks than the succeeding lines. Such paragraphs are handled by RISOE/BRAILLE using a negative value of submargin. Again the result is that the original paragraph structure is conserved, i.e. each Backus-Naur formula is printed as a series of Braille lines of which the first line has fewer leading blanks than the other lines.

It is of course decisive for the usefulness of the system described that options can change value in the middle of the text. RISOE/BRAILLE therefore allows the text to contain option cards i.e. records in which value of options are changed.

The Algol Manual was punched on cards with relation to Braille production and of course, the cards were punched directly as input to RISOE/BRAILLE. The machine-readable form of the Input/Output Manual has obviously been produced with relation to a version in normal print (output on the lineprinter), and it was therefore necessary to code a preprocessor which transforms the original text into a form that can be used as input to RISOE/BRAILLE. The most important task of the preprocessor is to analyse the paragraph structure of the original text, and based on this analysis to insert option cards in the text. The Braille versions of manuals thus produced are without abbreviations, and are therefore quite voluminous.

EBCDIC-Braille has been used instead of usual Braille, but as mentioned earlier this does not constitute any serious inconvenience. It should be emphasized that the Braille versions are exact copies of the versions in normal print, and thereby put the blind employee on an equal footing with his sighted colleagues.

In October 1978 a theme issue appeared of the periodical "Ingeniøren" ("The Engineer") with the theme "edp of everyday use". The blind employee contributed to this issue with an article on automatic Braille production. The periodical is printed by use of photo composition, and the machine-readable test which is thereby produced was used to print the issue in Braille. The main purpose was to demonstrate that machine-readable text may be transformed into Braille by an automatic process, even if the machine readable text is produced with no relation to Braille production. The text from the photo composition system (paper-tape) was read to the disc storage of the B6700. A preprocessor was coded which transformed the text so that it could be used as input to RISOE/BRAILLE.

As understood from the above the result is not perfect, but a fully useable Braille version of the periodical was produced.

4.7. The Future

As mentioned the hardware necessary for automatic production of Braille exists at the State Library for the Blind in Copenhagen. The Braille processor which the Danish Society of the Blind had produced is able to introduce abbreviations in the text with reasonable certainty.

The Braille processors that have been produced at RECKU and Risø have demonstrated that the editing problems may also be solved to a fair extend. In other words the most important elements in a system for automatic Braille production are already available.

In January 1979 a working group was set up whose main purpose is to investigate the possibilities of using all types of machine-readable text for the production of Braille and (secondarily) print in large type. Besides, the group is expected to suggest a production system. It appears as if the group has a very good chance of reaching a positive result.

5. RISØ COMPUTER LIBRARY

In 1976 an effort was started to reorganize the available application software at the B6700. It was decided at that time to build a library of basic routines covering general purpose subjects in applied mathematics including statistics and operations research, together with graphics, utilities etc. The programming language should be restricted to algol and fortran. There already existed locally developed routines (the SA- and SF-libraries) with roots back to the GIER computer; to these a large number of fortran subroutines from external libraries were added, and all this was put together to form Risø Computer Library (RCL) with several hundred items. A new comprehensive abstract list, Risø Computer Library Abstract (RCLA), was produced. With the installation of disk-pack storage it became feasible to maintain both RCL and RCLA as file collections in the computer, whereby updatings and corrections could easily be made. When possible, descriptions were made as comments within the symbolic program texts.

RCLA contain only few complete programs. Most of the units are subprograms intended to serve as fundamental bricks in the users' algol and fortran programs. Existing Risø algorithms for special purposes such as reactor physics calculations cannot be found in RCL; they are filed in the user libraries outside the Risø Computer Library.

The abstract list, RCLA, is organized by use of a modified SHARE classification index with certain local extensions. The first page of the index is shown in Fig. 5.1. RCL and RCLA form a dynamic system in the sense that the current version of RCLA always reflects the current status of RCL. Each time RCL has undergone a significant amount of change, new issues of RCLA are printed and sent to the departments of Risø; also some copies are forwarded to certain external Danish computing centers.

Since 1976 Risø Computer Library has undergone substantial changes in contents. Old locally produced procedures and subroutines from IBM's SSP-collection have been and are being replaced by algorithms satisfying modern standards. The Computer Installation is constantly looking for available high-quality routines or packages of routines. As an example, the EISPACK-2 collection of about 70 subroutines for linear eigenvalue problems was installed and succesfully tested in 1978. We also write some routines ourselves, either from scratch or by using an existing code as a model.

The abstract list RCLA is maintained and edited by special utility programs. Recently the list was split into an algol and a fortran part, and an alphabetic subject entry was added; after these changes the list is easier to use. Each item in RCLA contains the file name of the algorithm, a short abstract, and reference to a description (see Fig. 5.2).

At the moment all algorithms in the RCL are kept on the system-resource disk-pack which is permanently mounted. Back-up versions of RCL on tape are made regularly in order to protect them against loss due to system breakdowns.

It is our policy to restrict the Risø Computer Library to basic algorithms written in algol and fortran. Larger computing centers, in particular those associated with the universities, also offer comprehensive utility systems for say graphics, statistics, economy, structural analysis, text editing, data base management, etc.

Installation of such integrated systems will normally exceed the resources of the Computer Installation in manpower, in computer demand, or in licence fees. Also, we dount that Risø's users, who are engaged in research and development, could utilize such large scale systems to a reasonable extent. Probably it is more reasonable to refer sporadic demands of this kind to the University Computing Centers in the Copenhagen area.

The staff working on the algorithm project is very limited, and the same people also work in different projects in physics, applied mathematics, etc., in collaboration with various groups at Risø. We believe that this contact has a positive feedback to the algorithm project. New algorithms often come as spin-off from such projects. Also, we sometimes get a request from a user about some piece of mathematical software not found in RCL. Then we try to look for a suitable algorithm from the literature or we may contact institutes outside Risø, or we write the routine ourselves.

Like other information systems some units of RCL are used much more frequently than the rest. A few routines, like equation solvers, are used very often whereas others probably have not been used at all until now. We try, however, to foresee as far as we can in which areas there is a need for basic algorithms.

Instead of maintaining our own library we might have adopted some foreign program library, for example the British NAG library or the American IMSL library. Such program products are proprietary however. Although the license fee may be moderate, Risø users would not have the free right of disposal for programs using the licensed algorithms in their research projects; this could be an obstacle to exchanging programs with colleagues.

It has been our experience that it is more difficult to procure algol procedures of modern standard than fortran subroutines, because fortran is more widespread and better standardized than algol. Nevertheless we try to keep RCL up-to-date in both languages, and Risø is, as a matter of fact, one of the few places where a well-equipped algol procedure library exists.

Regarding the portability of Risø Computer Library, the fortran part is almost completely portable. This means for example that if the Risø B6700 should be replaced by a computer manufactured by someone else than Burroughs, it will be an easy task to install FORTRAN-RCL on such a machine. The algol part is not portable to the same extent as non-Burroughs equipment. This is due to inevitable differences in algol dialects on different computers, but we have endeavoured to keep the algol procedures close to ALGOL60.

Fig. 5.1. The first page of RCLA Alphabetic Classification Index.

```
E01   INTERPOLATION AND TABLE OPERATIONS                    ALGOL

RISOE/SPLINEVAL/A   B6700 ALGOL PROCEDURE SPLINEVAL          E01

   PURPOSE      EVALUATES A CUBIC INTERPOLATING SPLINE AT AN AR-
                BITRARY POINT. THE COEFFICIENT ARRAYS ARE SUP-
                POSED TO BE COMPUTED BY RISOE/CUBESPLINE/A.
   CREATED      28 MAR. 1979.
   DESCRIPTION COMMENTS IN ALGORITHM.

RISOE/CUBESPLINE/A  B6700 ALGOL PROCEDURE CUBESPLINE         E01

   PURPOSE      COMPUTES THE COEFFICIENT ARRAYS FOR A CUBIC
                INTERPOLATING SPLINE THROUGH A SET OF POINTS.
   CREATED      28 MAR. 1979.
   DESCRIPTION COMMENTS IN ALGORITHM.
   REMARKS      EVALUATION OF THE INTERPOLATING FUNCTION AT AN
                ARBITRARY POINT CAN BE DONE BY SUBSEQUENT USE OF
                RISOE/SPLINEVAL/A.
```

Fig. 5.2. Examples of abstracts.

6. ADAPTIVE NUMERICAL QUADRATURE

The computation of integrals appearing in physical and technical problems sometimes presents difficulties because it in many cases is not possible to obtain an analytical expression for the integral. In such cases one is thus reduced to apply numerical methods. The number of numerical quadrature methods is legion and we faced the problem to choose the overall most robust and efficient standard method to be included in the Risø Computer Library.

An often used method at Risø for one-dimensional problems is the well-known Romberg method which continues the equidistant subdivision of the given integration interval until the error has passed the accuracy test (if possible!). The application of the Romberg method will sometimes give a wrong answer and it often requires an enormous number of function evaluations. There was a need for a more economical and reliable standard quadrature method.

A way to ease the situation is to use a method which is self-adjusting to the problem at hand. Some methods are called adaptive. However, it must be borne in mind, that the "method" which solves all problems does not exist; every method - it might be as good as possible - can always be "fooled" by an intelligent choice of functions.

In order to choose a method which can evaluate the integral of function types which are frequently of interest, we started a thorough study of all adaptive as well as non-adaptive quadrature methods in current use which could be found in the literature on numerical integration methods together with unpublished methods obtained by personal contacts.

13 different methods were collected and tested systematically on Risø's B6700 computer with a set of 86 test functions of different complexity. Some of the functions have removable discontinuities, some have integrable singularities. The test

functions were for the greater part taken from papers concerned with numerical quadrature methods.

The common feature of the adaptive methods studied is that they use fixed-order quadrature rules, either a Simpson, a Gaussian or a Newton-Cotes rule. The methods make a subdivision of the integration interval in such a way that the subdivision is dense in those parts of the original interval where the function has its greatest variation. That leads of course to an optimal economy in the computation. What separates the different methods using the same type of rule is the strategy.

The result of the comparison of the different methods was very surprising. The method which was most promising in overall performance with respect to robustness against difficult functions (discontinuities and integrable singularities) and the number of function evaluations is an unpublished Danish method from the early sixties invented by A. Rieschel, A/S Regnecentralen and further developed by S.E. Christiansen, The Royal Veterinary and Agricultural University, Copenhagen.

Their method is based on application of a 5 and a modified 7-point Newton-Cotes quadrature rules. The 5-point rule is the usual one, but the 7-point rule is a special one as the two outermost division points in the actual subdivision are placed such that the further division of the subinterval gives a good computation economy. At the same time this point distribution is more similar to the 7-point Gaussian quadrature rule which leads to greater accuracy.

Our study of these quadrature methods has led us to dismiss the Romberg method used until now as the standard method in Risø Computer Library and to accept the new method which is inclosed in the library both as an algol procedure and a fortran subroutine.

7. THE NUMERICAL EVALUATION OF A SPECIAL 3-DIMENSIONAL INTEGRAL APPEARING IN THE THEORY OF THE GAUSSIAN DISPERSION MODEL

The computation of external γ-dose at a given point from a plume of radioactive material released from a stack, e.g. by a nuclear power plant is an important but difficult problem, even if the plume is stationary in time.

The received γ-dose at a point (XD,YD,ZD) is, to a good approximation, given by an expression in 3 variables x,y,z:

$$D_g = K_e \cdot T_e/(8\pi^2 u) \sum_{i=1}^{N_i} \sum_{k=1}^{N_k} \left[S_{i,k} \int_{x_1}^{x_2} \int_{-H}^{\infty} \int_{-\infty}^{\infty} F_1(x) \cdot F_2(x,z) \cdot F_3(x,z,y) \; dy \; dz \; dx \right]$$

where

$$S_{i,k} = \varepsilon_i \cdot G_k \cdot \sigma_k \cdot f_{k,i}$$

$$F_1(x) = \exp(-\lambda_i \cdot x/u) \sigma_y^{-1}(x) \sigma_x^{-1}(x)$$

$$F_2(x,z) = \exp(-z^2/2\sigma_z^2(x)) + \exp(-(z+2H)^2/(2\sigma_z^2(x)))$$

$$F_3(x,z,y) = (1+K_k \cdot \mu_k \cdot r)/r^2 \cdot \exp(-\mu_k r) \cdot \exp(-y^2/(2\sigma_y^{\;2}(x)))$$

$$r^2 = (x-XD)^2 + (y-YD)^2 + (z-ZD)^2$$

The cartesian coordinate system is placed with origon at the top of the stack, x-axis pointing downwind from the stack and z-axis pointing upward perpendicular to the ground.

u is the wind speed (in the x direction) $\sigma_y(x)$ and $\sigma_z(x)$ are the dispersions in y and z directions and are given as tabulated functions of x. r is the distance from the point (XD,YD,ZD) to a point (x,y,z) in the plume, and H is the height of the stack. The remaining letters with or without subscripts are constants.

It is evident that the expression for D_g must be difficult to evaluate just because of the exponentials which are dependent on functions of x. Experiments of straightforward application of the product form of Gaussian quadrature rules lead to non-acceptable long computing times, and what may be wanted in this case - a standard adaptive numerical quadrature method in three dimensions - does not exist at the present time.

To solve the problems we have constructed weight functions based on the exponentials and the function $1/r^2$ and cut off a thin slice in the integration domain around the point (XD,YD,ZD) to prevent the singularity for r going to zero.

The implementation of this procedure leads to a program which performs the integration in an acceptable short time.

The method has been developed for the Health Physics Department to be used in calculating the distribution of individual and collective γ-doses in the area surrounding a nuclear installation.

The importance of short computing times is also evident if the program shall be used in connection with estimating possible γ-doses as a result of an accidental release of radioactive materials.

8. SIMULATION OF CHEMICAL REACTION SYSTEMS

In collaboration with the Accelerator Department a program for simulation of chemical reaction systems has been developed. The work was started some years ago and has reached its final form in 1978. It has proved to be a valuable tool for the chemists at Risø working in reaction kinetics.

The purpose of the program is to study the development in time of selected chemical reaction systems. This program is particularly suited for systems which are subjected to radiation ranging from a short time intense radiation, the so-called "pulse", to a long term "weak" radiation.

The study of chemical kinetics involves the numerical solution of the corresponding nonlinear ordinary differential equation systems. It turns out that the changes of the concentrations of the different chemical reactants at a given time are governed by time constants which can vary by several orders of magnitude. This means that the differential equation systems are of the stiff stable type.

In order to solve such systems the numerical method devised by C.E. Gear has been adopted.

The program has an input routine which accepts the chemical reaction equations as input for a translation to the corresponding differential equation system.

The program is roughly divided in four parts:

1. An input part for all relevant start data such as the chemical reaction equations, starting concentrations of the reactants, reaction constants, relative accuracy of the integration, the time interval for the integration, etc. In the case of radiation the start data also contain the G-values which are the amount of reactants produced by radiation.
2. The translation part which translates the reaction system to the corresponding differential equation system and controls the stoichiometry of the reaction system. If the stoichiometry is not correct or if the user of the program has not followed the input conventions for the input of data, error messages are returned from the program.
3. The numerical integration part which performs the computations.

4. The output part which prints all the desired results including graphs showing time variations of selected reactants. The limit for the chemical reaction system to be simulated is given by the size of the computer alone, i.e. no practical limits exist for most computers in hand today.

The program which is written in B6700 extended algol has been adopted by the chemists at Risø as a very useful tool in studying chemical kinetics. One interesting application of the program among others is the simulation of chemical reaction kinetics in connection with the deposition of radioactive waste products.

On the next 4 pages an example is shown of a computation for a simple system, which describes the initial reaction in CU[++]/O_2-system.

8.1. Example

```
NEW REACTION EQUATION SYSTEM
RE1:E[-]+CU[++]=CU[+];3@10
RE2:H+CU[++]=CU[+]+H[+];2@8
RE4:O2[-]+H[+]=HO2;2@11
RE5:H+O2=HO2;2@10
RE6:E[-]+H[+]=H;2.3@10
NEW G - VALUES
G(E[-])=2.95
G(H)=0.55
NEW START CONCENTRATIONS
CON(CU[++])=@-2
CON(O2)=1.2@-3
CON(H[+])=@-2
NEW PRINTING EXPRESSIONS
PE1:E[-];
PE2:CU[+];
PE3:H;

                    DIFFERENTIAL-EQUATION SYSTEM


D[E[-]]/DT   = - K1*[E[-]]*[CU[++]] - K5*[E[-]]*[H[+]]
               + G(E[-])*CONST*DOSE


D[CU[++]]/DT= - K1*[E[-]]*[CU[++]] - K2*[H]*[CU[++]]


D[CU[+]]/DT = + K1*[E[-]]*[CU[++]] + K2*[H]*[CU[++]]


D[H]/DT      = - K2*[H]*[CU[++]] - K4*[H]*[O2] + K5*[E[-]]*[H[+]]
               + G(H)*CONST*DOSE


D[H[+]]/DT   = + K2*[H]*[CU[++]] - K3*[O2[-]]*[H[+]]
               - K5*[E[-]]*[H[+]]


D[O2[-]]/DT = - K3*[O2[-]]*[H[+]]


D[HO2]/DT    = + K3*[O2[-]]*[H[+]] + K4*[H]*[O2]


D[O2]/DT     = - K4*[H]*[O2]

THE VALUES OF THE REACTION CONSTANTS

K1 = 3.000E+10  K2 = 2.000E+08  K3 = 2.000E+11  K4 = 2.000E+10
K5 = 2.300E+10


G(E[-])   = 2.950          CON(E[-])   = 0.
G(CU[++])= 0.000          CON(CU[++])= 1.000E-02
G(CU[+]) = 0.000          CON(CU[+]) = 0.
G(H)      = 0.550          CON(H)      = 0.
G(H[+])   = 0.000          CON(H[+])   = 1.000E-02
G(O2[-]) = 0.000          CON(O2[-]) = 0.
G(HO2)    = 0.000          CON(HO2)    = 0.
G(O2)     = 0.000          CON(O2)     = 1.200E-03
```

```
TOTAL DOSE.......................= 9.64E-01  KILORAD
NUMBER OF RADIATIONS.............= 1
RADIATION-TIME...................= 1.0E-12 SEC.
INITIAL INTEGRATION STEP.........= 1.0E-13 SEC.
NO. OF RESULTS DURING RARIATION..= 10
TOTAL NO. OF RESULTS.............= 30
MAXIMUM INTEGRATION TIME.........= 1.0E-03 SEC.
RELATIVE ACCURACY IN DIFSUB......= 1.0E-05
MINIMUM-INTEGRATION-STEP.........= 1.0E-14
MAXIMUM-INTEGRATION-STEP.........= 2.0E-05
```

METHOD OF INTEGRATION IS THAT OF C.W.GEAR FOR STIFF SYSTEMS.

RESULTS IN : LOG. TIMESCALE, LIN. Y-SCALE

PULSENUMBER= 1

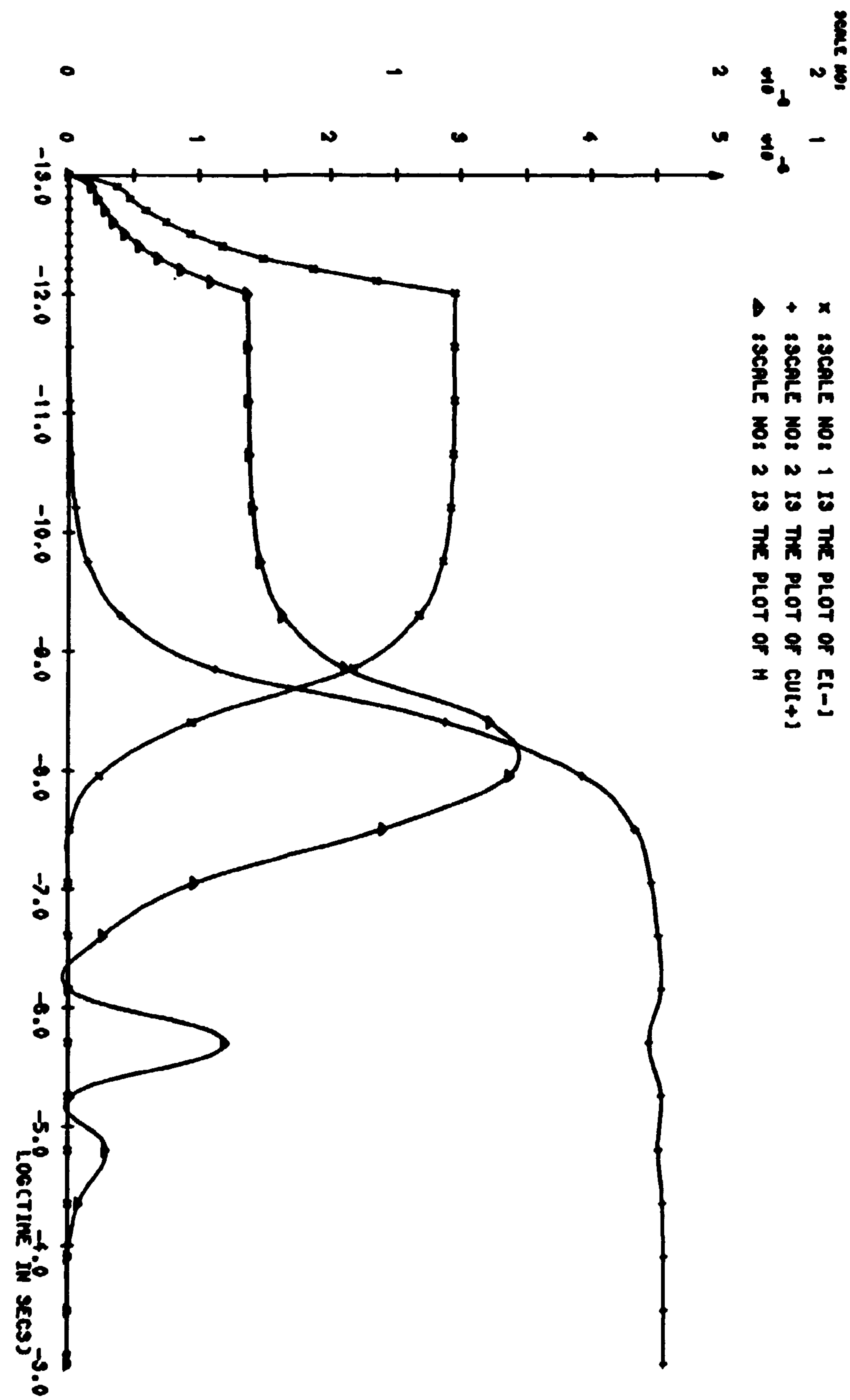
TIME	E[-]	CU[++]	CU[+]	H
0.	0.	1.000E-02	0.	0.
1.25893E-13	3.716E-07	1.000E-02	1.589E-11	6.930E-08
1.58489E-13	4.678E-07	1.000E-02	2.000E-11	8.724E-08
1.99526E-13	5.889E-07	1.000E-02	2.651E-11	1.098E-07
2.51189E-13	7.417E-07	1.000E-02	0.	1.381E-07
3.16228E-13	9.340E-07	1.000E-02	0.	1.738E-07
3.98107E-13	1.176E-06	1.000E-02	0.	2.186E-07
5.01187E-13	1.481E-06	1.000E-02	0.	2.751E-07
6.30957E-13	1.865E-06	1.000E-02	0.	3.462E-07
7.94328E-13	2.348E-06	1.000E-02	0.	4.358E-07
1.00000E-12	2.956E-06	1.000E-02	0.	5.486E-07
2.81838E-12	2.953E-06	1.000E-02	1.489E-09	5.497E-07
7.94328E-12	2.950E-06	1.000E-02	3.114E-09	5.508E-07
2.23872E-11	2.942E-06	1.000E-02	7.695E-09	5.538E-07
6.30957E-11	2.920E-06	1.000E-02	2.061E-08	5.622E-07
1.77828E-10	2.856E-06	1.000E-02	5.700E-08	5.862E-07
5.01187E-10	2.676E-06	1.000E-02	1.596E-07	6.535E-07
1.41254E-09	2.169E-06	1.000E-02	4.486E-07	8.434E-07
3.98107E-09	9.398E-07	9.999E-03	1.151E-06	1.290E-06
1.12202E-08	2.316E-07	9.998E-03	1.571E-06	1.349E-06
3.16228E-08	9.099E-09	9.998E-03	1.735E-06	9.612E-07
8.91251E-08	2.943E-10	9.998E-03	1.784E-06	3.876E-07
2.51189E-07	6.261E-12	9.998E-03	1.806E-06	1.059E-07
7.07946E-07	6.600E-16	9.998E-03	1.814E-06	2.446E-09
1.99526E-06	2.182E-12	9.998E-03	1.778E-06	4.800E-07
5.62341E-06	0.	9.998E-03	1.814E-06	9.744E-09
1.58489E-05	0.	9.998E-03	1.806E-06	1.158E-07
4.46684E-05	0.	9.998E-03	1.820E-06	3.378E-08
1.25893E-04	0.	9.998E-03	1.823E-06	0.
3.54813E-04	0.	9.998E-03	1.823E-06	0.
1.00000E-03	0.	9.998E-03	1.823E-06	0.

TIME	H[+]	O2[-]	HO2	O2
0.	1.000E-02	0.	0.	1.200E-03
1.25893E-13	1.000E-02	0.	2.368E-13	1.200E-03
1.58489E-13	1.000E-02	0.	2.980E-13	1.200E-03
1.99526E-13	1.000E-02	0.	3.951E-13	1.200E-03
2.51189E-13	1.000E-02	0.	6.073E-12	1.200E-03
3.16228E-13	1.000E-02	0.	1.334E-11	1.200E-03
3.98107E-13	1.000E-02	0.	2.257E-11	1.200E-03
5.01187E-13	1.000E-02	0.	3.433E-11	1.200E-03
6.30957E-13	1.000E-02	0.	4.935E-11	1.200E-03
7.94328E-13	1.000E-02	0.	6.860E-11	1.200E-03
1.00000E-12	1.000E-02	0.	9.338E-11	1.200E-03
2.81838E-12	1.000E-02	0.	1.211E-10	1.200E-03
7.94328E-12	1.000E-02	0.	2.765E-10	1.200E-03
2.23872E-11	1.000E-02	0.	7.145E-10	1.200E-03
6.30957E-11	1.000E-02	0.	1.949E-09	1.200E-03
1.77828E-10	1.000E-02	0.	5.428E-09	1.200E-03
5.01187E-10	1.000E-02	0.	1.523E-08	1.200E-03
1.41254E-09	1.000E-02	0.	4.287E-08	1.200E-03
3.98107E-09	9.999E-03	0.	1.230E-07	1.200E-03
1.12202E-08	9.999E-03	0.	3.525E-07	1.200E-03
3.16228E-08	9.999E-03	0.	7.996E-07	1.199E-03
8.91251E-08	9.999E-03	0.	1.332E-06	1.199E-03
2.51189E-07	9.999E-03	0.	1.593E-06	1.198E-03
7.07946E-07	9.999E-03	0.	1.688E-06	1.198E-03
1.99526E-06	9.999E-03	0.	1.247E-06	1.199E-03
5.62341E-06	9.999E-03	0.	1.680E-06	1.198E-03
1.58489E-05	9.999E-03	0.	1.582E-06	1.198E-03
4.46684E-05	9.999E-03	0.	1.651E-06	1.198E-03
1.25893E-04	9.999E-03	0.	1.681E-06	1.198E-03
3.54813E-04	9.999E-03	0.	1.681E-06	1.198E-03
1.00000E-03	9.999E-03	0.	1.681E-06	1.198E-03

NO. OF INTEGRATIONSTEPS= 102
MAXIMUMSTEP= 2.000E-05
MINIMUMSTEP= 1.000E-14
COMP. TIME= 12.55 SEC.

END OF THIS COMPUTATION

****



Graph showing the concentrations as function of time for the chemical reactants E^-, CU^+, and H.

9. SOME MINOR PROJECTS

During the year the Computer Installation has worked on many projects that are too small to justify a separate paper in this report. A considerable part of these activities are the daily operations, the ongoing assistance to users, and formal and informal representation of Risø National Laboratory in computer matters, but a few deserve to be mentioned.

9.1. Additions to RCL

This report contains a paper with a general discussion of RCL, the Risø Computer Library. What has been added to RCL in 1978 either to replace older routines or to cover new subjects is specifically listed below.

Subject	Language (A=algol, F=fortran)
Linear eigenvalue problems: EISPACK, second release.	F
Adaptive numerical quadrature (see paper on this subject).	A+F
Positive-definite bandstructured linear equation systems (requested by the Engineering Department).	F
Complex linear equations.	F
Selected special functions: conversion of FUNCPACK2 from CDC-code to B6700.	F
Simple statistical sample analysis (tally).	F

Subject	Language (A=algol, F=fortran)
Statistical distribution functions: normal, inverse normal, t-, F-, and χ^2-distribution.	F
Analysis of variance, one-way.	F
Cubic spline interpolation.	F
Internal numerical sorting: van Emden's quicksort algorithm.	A+F

9.2. Computerization of a Model for Energy Supply in Denmark

A program for maintenance of a file containing data about the Danish energy supply network has been constructed for the Energy Systems Groups.

The network is modelled as a number of connected sources, lines, and drains. All items within the file are referenced by name from an interactive terminal program.

The program can create, delete, modify and display items within the file.

The maintenance of the file is done by the GET-PUT routines mentioned below in the paragraph "Simple Data-base".

9.3. Data Transmission to the Regional Centres

Risø is using the computer installations at the University and the Technical University of Copenhagen. One of the ways to access these machines is to let the B6700 computer simulate a Remote Job Entry (RJE) terminal. The RJE simulation program for the Univac 1100 at the University of Copenhagen has been rewritten to simulate a NTR terminal instead of the previous

DCT 1000 terminal. The change was forced by a system exchange at the University of Copenhagen. However, the modification also caused a throughput improvement by a factor of three on the telephone line.

9.4. Exchange of Files on Magnetic Tapes

Two utilities have been implemented to handle exchange of files on magnetic tapes from and to other computer installations. Files are sent to other installations by transferring an optional number of files from disk to unlabelled tape, with full documentation of contents, i.e., a summary of tape attribute specifications and individual file title, recordsize, and blocksize attributes followed by full listings of each file. Only the file titles are input to the utility. Receiving files from other installations is performed by transferring all or selected files from the tape to disk. Specification of file title, recordsize, and blocksize attributes are input to the utility.

Receiving magnetic tapes is often a troublesome subject. Documentation on tape formats and contents may be insufficient or in error. Tape formats may be incompatible with the local software and hardware capabilities. Hardware incompatibilities require tape transformation at another installation. Software incompatibilities require special actions involving datatransformations after transfer or reprogramming of a utility to perform even the initial data transfer from tape to disk.

9.5. Programs for Nuclear Geophysics

The program complex GAMP1/GFX, which solves the plane infinite two-media transport problem for terrestrial gamma radiation, was redesigned using an improved integration technique for the energy variable. The new method is described in a paper submitted for publication in "Journal of Computational Physics". Also redesigned were the auxiliary data handling programs to use direct-access disk-pack files rather than tape files.

The corresponding problem, where gamma flux or dose from a finite source is asked for, has also been studied. Due to the geometric complexity one must here resort to Monte Carlo methods. The adjoint Monte Carlo program GAMO was extended and tested, and a new code GMC based on forward Monte Carlo was written.

The computer models mentioned above are developed as a joint project with the Electronics Department. They are intented to serve as tools for various assessments in airborne radiometric survey and in calibration of gamma-ray spectrometers.

9.6. Simple Data Base with Named Data Items

A simple subset of data base facilities has been implemented in algol and fortran. Data-access is obtained by reference of named items. Users are relieved from considerations of data locations, dynamic dimensioning of individual data items, and input-output.

The algol version has been improved by making the internal dictionary core resident, which has reduced search time significantly.

9.7. Transmission of Binary Data on Paper Tape

Two utilities have been implemented to interface datatransmission of binary data from and to paper tape through a PDP-8, using standard B6700-Cande tape request, (TTY).

The binary data are transformed into hexadecimal representation before being transmitted. The recipient interpretes the hexadecimal data and transforms back to binary representation. The transformation is necessary to avoid accidental coincidence of control character patterns.

10. PLANNING RISØ's FUTURE COMPUTER INSTALLATION

Planning the future is a continuous activity, although it is often little explicit. However, we have reached a point in the life cycle of our present installation where a considerable effort during the year has been put into formal long range planning, i.e. until the mid-to-late eighties. The basic philosophies and conclusions of this planning are summarized in this paper.

Ideally, long range planning is a self-timing activity, but in a pragmatic world it often must be coupled to the economic realities. An optimal plan, calling for an investment in short time, may be discarded for another plan where the investment is postponed and extra expenses are needed to cover the time until then.

As a matter of fact two different plans have been worked on during 1978. The first plan called for a considerable growth in computer power in late '79. However, that plan was put aside and it was decided to extend the B6700 with an extra central processor in early 1979, and a second plan puts the radical change in early '82.

Nevertheless, the basic situation and philosophies in the two plans are the same, and they are the subject of this paper. It should be noticed that the plan will be the basis for an appropriation, not a proposal to buy, lease or rent a specified configuration.

The contents of the plan are therefore

- prognosis for the need of resources
- discussion of different methods to fulfil the need and the associated costs
- conclusions and proposal for future activities.

10.1. Prognosis

We have considered three different bases for a prognosis:

- extrapolation of the history
- comparison with similar institutions
- specific knowledge or expectation about future work.

Our experience shows that central processing power is the key resource in Risø's use of edp. Any mainframe computer with sufficient processing power can provide the necessary input-output channels, line printers, disk packs, data communication, etc.

It is typical for our scientific users that they want computing power when it fits their activities, not after a preplanned, regular schedule. This means that the load on the computer fluctuates on a daily, weekly, and monthly basis. However, on a yearly basis there is a definite growth in the use of the B6700.

During the years from 1972 to '78 the number of CPU-hours has grown between 9% and 46% from one year to the next. The total growth in this period equals an exponential growth of 21% per year.

In addition to this, Risø has access to several external computers. When this capacity is added, the total growth has been 25% per year in the period mentioned.

It is common knowledge that the use of edp has grown extremely quickly in all research institutions since the day edp was first introduced, although there is some correlation with the general, economic development. Examples from USA, Germany, and Denmark indicate that the above-mentioned 25% per year is a rather low figure for institutions of the same size as Risø.

In the summer 1977 a score of the more important user groups were interviewed about their plans for their future use of edp. Generally speaking, actual plans appeared to be restricted to a scope of 1-3 years corresponding to current projects. Which research projects will be considered promising three years from now are unknown, and in particular the postponement of a political decision on nuclear power in Denmark makes a precise calculation of future need impossible.

Fortunately, a precise knowledge of future work is not necessary in order to make plans for edp. We feel it safe to maintain that within the overall framework of Risø, the size of the staff has more influence on the necessary computing power than the detailed projects.

It is our estimate that a reasonable, harmonic growth in computer use will be between 20% and 30% a year for some years to come. Important political decisions, e.g. building of nuclear power stations in Denmark, may produce a situation where existing programs come into heavy use and the abovementioned growth figures will rise sharply.

The proper way to handle this possibility in the planning is to introduce the potential of large, compatible extension of the computer as an important criterion of the decision.

10.2. Methods to Fulfil the Need

From a technical point of view three different methods to provide additional edp-capacity are in clear focus: distributed processing power in the form of minicomputers, a central main frame at Risø, and terminals connected to larger installations outside Risø. The problem is to find a reasonable mixture of these methods.

Our surveys show that the use of external capacity is more expensive than the capacity of an on-site main frame of a suitable size. Furthermore the on-site installation serves as the centre of data-processing know-how, which it is impossible to maintain without such an installation. The daily operations and handling of sudden, high-priority jobs are more flexible, and security problems are smaller.

All these points indicate the value of an on-site main frame. However, there will always be some jobs which benefit from being executed externally, either due to their size or because the program is adjusted to a specific computer. External capacity is also a useful help in periods with heavy load, e.g. in periods shortly before new extensions of the main frame.

A study has shown that between 70 and 80% of our capacity is used for jobs which are too big to be run in a minicomputer. Admittedly, minies grow but so do our jobs, and the percentage mentioned will probably not change significantly during the next years. Therefore, the use of minies is not a viable alternative to a main frame.

The other 20-30% of the use could be moved to a number of minies, but this solution is expensive, mainly because either the minies will have very much idle time, or they will need an operating organisation to be shared between many users.

It should be borne in mind that the characteristics of the computer-use in a research laboratory is different than of commercial data processing, and the current philosophies about distributed data processing do not apply very well to our situation.

However, there are and will be projects at Risø which can benefit from having a minicomputer at their exclusive disposal and such solutions should not be barred in cases where the extra expense is considered justified. However, these cases do not sum up to a level that influences the necessary size of the main frame.

One more aspect to consider is that nearly all programming and testing is performed by the end-users themselves as an integral part of their research work. Data processing is "a necessary evil" to them, and to move programs between non-compatible systems, not to mention from one programming language to another, would be a heavy, unproductive, and meaningless burden.

Therefore, it is of great value to provide a system that, to as many users as possible, can be the only system they need to know about. This is true, both at a certain point in time and in a longer span of time which means that radical changes in the computing facilities are done better in few large steps than in many small ones.

10.3. Conclusions and Proposal

The conclusion of the performed planning has been that Risø shall stay with the edp-structure consisting of a central mainframe and a net of interactive terminals, some minicomputers and access to larger, external computing centres.

The number of terminals, in particular of graphic terminals, is expected to grow quickly, and we will need one or two more remote batch terminals.

Economical considerations point to the year 1982 for a major change. At that time our present B6700 will be so old that further extensions, e.g. with a back-end number cruncher processor, are unrealistic. Therefore, the plan calls for a replacement. The capacity of the new mainframe should be about twice the load when the replacement takes place. In this way we get a surplus capacity in the beginning to absorb errors during a learning period and a reasonable period before new extensions.

A number of extensions - up to about five times the initial capacity - is foreseen through the lifetime of this computer. They will be either speeded up or delayed to compensate for differences between the present prognosis and the actual development.

The obvious thing to do is to buy a larger Burroughs computer. However, the obvious is not necessarily the best, so we intend to ask for offers from any vendor who wants to bid. In the evaluation of the offers we must of course estimate the conversion expense of all existing programs and data and take these expenses into consideration when the offers are compared.

APPENDIX A

CONFIGURATION OF RISØ's B6700 COMPUTER AS OF DECEMBER 1978

1	Central Processor, 5 MHz.
1	Input-Output Processor with 6 floating channels.
48	K word ferrite core, 51 bit plus 1 parity bit, cycle time 1.2 μs.
192	K word ferrite core, 51 bit plus 9 parity bit, cycle time 1.6 μs.
3	Modules Head per Track Disk File, each 1.7 M words.
1	Disk Storage/Dual Controller and 1 Dual Drive Increment with a total of 4 Disk Drives, 14.6 M words per Disk Pack.
1	Magnetic Tape Cluster with 4 Tape Transports, 9 tracks, 800 bpi, 36 KB.
1	Card Reader, 800 cards per minute.
1	Card Punch, 100 cards per minute.
1	Paper Tape Reader, 1000 char/sec.
1	Paper Tape Punch, 100 char/sec.
1	Line Printer, 815 lines per minute, 132 print positions, 64 char.
1	Operator Console Display.
1	Data Communications Processor, 4 k local storage.
12	Line Adapters with Dial-up 300 baud Modems for Visual Display Units and Printing Terminals.
1	Line Adapter with Dial-up 1200 baud Modem for special Terminals.
1	Line Adapter, directly connected to a VDU.
2	Line Adapters, for PDP-8 Minicomputers.
2	Line Adapters with locally built interface for 2 Digital Plotters (Calcomp 507 and 565).
1	Lineadapter with DATEL 2400 Modem for external Computers.
1	Line Adapter for Remote Job Entry Terminal.

APPENDIX B

STAFF OF THE COMPUTER INSTALLATION

Head: Leif Hansson

Office staff: Lill Hansen
Vivi Holm
Birthe Jessen (until February 15)

Scientific staff

Erik Hansen
Peter Kirkegaard
O. Lang Rasmussen
Preben Voss

Technical staff

Carl Bergman
Birthe Berland*
Hanne Bundgård*
Jørn Deutschbein
Preben Folkjær*
Søren Frederiksen
Ib Hansen
Helle Kiærulf Kristensen
Steen Rahbek Petersen
Henriette Rasmussen

---

* Temporary assistant

APPENDIX C

	Production	On-line	PM	UM	Processor
January 78	414.25	1075.35	33.30	9.00	213.03
February	435.20	1450.12	28.00	17.30	291.36
March	393.25	1127.38	19.30	100.05	264.21
April	358.40	1006.33	28.00	12.00	247.02
May	390.50	1196.04	21.00	8.30	226.16
June	463.00	1390.29	20.30	24.30	305.52
July	440.05	1012.54	33.0	0	275.57
August	533.20	1329.10	28.45	0	363.57
September	440.10	1315.59	25.00	0	320.56
October	376.15	1153.17	32.15	2.45	190.07
November	463.15	1521.24	28.30	4.45	316.02
December	349.40	1127.30	21.45	2.05	228.00

Production: Hours and minutes the computer has been on for production runs.

On-line: Sum of use of terminals, hours and minutes.

PM: Preventive Maintenance, hours and minutes.

UM: Unforeseen Maintenance including waiting time, hours and minutes.

Processor: Automatically logged use of central processor, hours and minutes.